UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/811,637	03/19/2001	Otto Lodewijk Steinbusch	NL 000113	9924

24737 7590 06/04/2004

PHILIPS INTELLECTUAL PROPERTY & STANDARDS
P.O. BOX 3001
BRIARCLIFF MANOR, NY 10510

EXAMINER	
VU, TUAN A	

ART UNIT	PAPER NUMBER
2124	6

DATE MAILED: 06/04/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/811,637	STEINBUSCH, OTTO LODEWIJK	
	Examiner	Art Unit	
	Tuan A Vu	2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *19 March 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-9* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-9* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *19 March 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *4*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the application filed March 19, 2001.

Claims 1-9 have been submitted for examination.

Information Disclosure Statement

2. The information disclosure statement (IDS) filed 3/19/2001 fails to comply with 37 CFR 1.98(a)(3) because it does not include a concise explanation of the relevance in regard to some content included, as it is presently understood by the individual designated in 37 CFR 1.56(c) most knowledgeable about the content of the information. It has been placed in the application file and very superficially and reluctantly considered because it appears that the subject matter presented therein exhibits no relevance to the claimed invention.

In fact, regarding the foreign patent document presented in the IDS, patent # EP 0009946 A1, the content is about an architecture of an X-Ray tube. It is urged that in order for the document to be fully considered Applicant provide appropriate explanation as to how the subject matter of said document relates to the present invention.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Darlet et al., USPN: 6,542,167 (hereinafter Darlet), in view of Gee et al., USPN: 6,317,872 (hereinafter Gee), and further in view of Lai et al., USPN: 6,382,846 (hereinafter Lai).

As per claim 1, Darlet discloses a method of executing a program, the program containing a particular instruction with a particular symbolic reference to operand data, the method comprising:

providing a symbol table containing information for resolving symbolic references in instructions (Fig. 3) from any category;

providing groups of memory locations, each group for storing symbolic reference-result associations, the result of the association having resulted from reference resolving for an instruction of the to which the group is assigned (e.g. structure 310 – col. 5, line 54 to col. 6, line 7; Fig. 5 – Note: grouping resolution result by module reads on group of memories storing symbolic-reference-result);

executing the particular instruction with consulting a memory group to determine whether a group containing a association for the particular symbolic reference and if there is such an association, using the result for that symbolic reference (e.g. col. 6, line 50 to col. 7, line 20); and using the symbol table to resolve unresolved references (e.g. Fig. 7)

But Darlet does not disclose supporting mutually different categories of instruction types; nor does Darlet disclose grouping the memory locations, each according to the category of instructions; nor storing the association with the symbolic reference result for an instruction of the category to which a group is assigned. Darlet only shows grouping memory into modules of code (Fig. 6) and associate symbol-reference/result offset value according to the modules (Fig. 5). Gee, in a method to organize the memory into structures analogous to modules by Darlet, discloses class static areas (CSA) which include indexed areas for storing offsets to reference resolution values; and classify these offset values into categories such as class or method or

instance field (e.g col. 13, line 18 to col. 14, line 46; Fig. 3). Gee's indexing is reminiscent of categorizing data in group to support fast search; and class, method name or instance field would be analogous to mutually supporting instruction categories, since instance field, like operand data, or method or class names are like numerical or symbolic representation of an instruction to be loaded or stack-based operations for Java runtime. To enhance the memory grouping as suggested by Darlet when memory resources are available, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify the association reference/result structure as taught by Darlet so to implement thereto the grouping as taught by Gee, such grouping associating reference symbols (of class, instance field, or method name) with symbol reference/resolution result, i.e. grouping into Java instruction categories. The motivation would be that in associating a symbol reference result of a particular category of operand data, e.g. method name, class name, or instantiated field, the execution time would be enhanced because the way the results are grouped into a form of index based on same family, e.g. class or method, of instruction symbol format would expedite the linking time of the execution environment by virtue of the well-known technique of implementing grouping via index as taught by Gee for search expediency.

Nor does Darlet disclose executing the instruction such that if there no association symbol reference/result in the above categories-based group in memory locations, resolving the particular reference, using the result from such resolving as operand data for executing the instruction and storing an association symbolic reference/ result of said resolving step in the group assigned to the category of the particular instruction. But in view of the teachings by Darlet (e.g. step 256 - Fig. 7; Fig. 5) and the storing of reference result in terms of offset values

in the indexed structures by Gee from above, the limitation of storing an symbol reference/result in a memory group would have been obvious for the same rationale as set forth above. Further, Lai, in a method to provide dynamic symbolic reference resolution analogous to Darlet's linking method, discloses index-based storing of the result in a numeric reference table/buffer (buffer *58-59* – Fig. 4; table *60* – Fig. 5), thus enabling memory storage of resolution numeric values to be stored for fast subsequent retrieval (col. 11, lines 24-45) analogous to Gee, and further teaches in case the numeric operand is not available, search for object data for obtaining the correct numeric operand and substitute it into the instruction (e.g. Fig. 6). It would have been obvious for one of ordinary skill in the art at the time the invention was made to implement the dynamic search, resolving and storing of the resulting operand numeric value by Lai and implement to combined Darlet/Gee's method for the necessary reason that without such dynamic resolution runtime linking would result in memory reference conflict; especially when the numeric value is stored in memory and ready for fast substitution in the instruction at runtime.

As per claim 2, Darlet does not disclose first and second instruction type that belong to one of the respective categories, each instruction require a common type of reference, such reference is different from that of another instruction belonging to another categories. But in view of Gee's categorizing of reference results according to mutually supporting Java class, method or instance field, such the reference in a group being of a particular type within such group/category and different from another reference type for another instruction belonging to another category is implicitly disclosed; because it is known that Java language would ensure that a class and inherent methods are defined the same and when instantiated object symbol reference are derived from the same class, they would necessarily fall under the same type as

opposed to another symbol being defined in another distinct class as guaranteed by the Java programming language.

As per claim 3, Darlet (combined with Gee/Lai) does teach that a group result comprise an information identifying an operand value (*Reference offsets* – Fig. 5); but does not including a data size indicator. Gee disclose object derived from analyzing a symbol table and including a *object_length* in building such object-related structure for storing the resolved access information (Fig. 13; col. 22, line 46 to col. 23, line 20); and Lai teaches substituting a resolved reference numeric value in the instruction under dynamic linking (*S20* - Fig. 6). In view of the need to substitute a given length of data into a field of the instruction being fetched for execution as taught by Lai, and in combination with the teaching to provide object length in resolution result structure, it would have been obvious for one of ordinary skill in the art at the time the invention was made to add the data size indicator in the association of reference/result as suggested by Darlet using the teachings by Gee and Lai, because the size information thus imparted to the association result as suggested by Darlet/Gee would provide exact memory allocation information to Lai's (combined with Darlet) method so that when data are replacing an instruction field, there would be better chance as to avert operand or field size mismatch when modifying a runtime instruction.

As per claim 4, Gee discloses a Java virtual machine (e.g. col. 3, line 45-61; Java Embedded Processor - Fig. 1 and related text). It is well-known that using of virtual machine for translating instruction into native method and executing the instructions resulting therefrom was a well-known concept in the art of programming; hence the limitation of having processor with

native instruction set and virtual machine instruction set is implicitly disclosed in the combination Darlet/Gee.

As per claim 5, Darlet and Gee teaches offset to reference values, hence have disclosed implicitly replacing the particular instruction loaded from runtime memory; further Lai teaches substituting numeric value, i.e. operand, into the instruction for execution and executing it instead of the original instruction being loaded/fetched prior to the resolving (re claim 1); hence the limitation is disclosed by virtue of the rationale to combine Darlet/Gee with Lai in claim 1.

As per claim 6, Darlet discloses a computer program product with a linking module to execute program instructions, such program arranged to:

providing a symbol table containing information for resolving symbolic references in instructions (Fig. 3) from any category;

providing groups of memory locations, each group for storing symbolic reference-result associations, the result of the association having resulted from reference resolving for an instruction of the to which the group is assigned (e.g. structure 310 – col. 5, line 54 to col. 6, line 7; Fig. 5 – Note: grouping resolution result by module reads on group of memories storing symbolic-reference-result);

executing the particular instruction with consulting a memory group to determine whether a group containing a association for the particular symbolic reference and if there is such an association, using the result for that symbolic reference (e.g. col. 6, line 50 to col. 7, line 20); and using the symbol table to resolve unresolved references (e.g. Fig. 7).

But Darlet does not teach executing native instructions translated from a virtual machine. At the time the invention was made, the use of Java or object-oriented language in conjunction

with virtual machine was a well-known concept in the art of code implementation. Gee, in a method to organize memory in structures to facilitate the execution of Java code by modification of program instructions based on information from those structures, teaches a virtual machine (col. 3, line 45-61; Java Embedded Processor - Fig. 1). The limitation as to provide a computer product by Darlet with a virtual machine as taught by Gee would have been obvious because it would allow Java programming and object-oriented benefits to Darlet's teachings and the inherent properties to using Java language, e.g. inheritance, encapsulation, portability, etc.

But Darlet does not disclose supporting mutually different categories of virtual machine instruction types; nor does Darlet disclose grouping the memory locations, each according to the category of instructions; nor storing the symbolic reference result for an instruction of the category to which a group is assigned. Nor does Darlet disclose executing the instruction such that if there no association symbol reference/result in the above categories-based group in memory locations, resolving the particular reference, using the result from such resolving as operand data for executing the instruction and storing an association symbolic reference/ result of said resolving step in the group assigned to the category of the particular instruction.

But these limitations have been addressed in claim 1 using Gee and Lai.

As per claim 7, refer to rationale of claim 2.

As per claim 8, Darlet discloses a data processing device for executing instructions, such device comprising

a first storage space, second storage units and execution unit for, respectively:

storing symbol table containing information for symbol reference resolving;

for storing symbolic reference-result associations;

for executing a particular instruction containing symbolic reference;

all of these steps being addressed with the corresponding rejections of claim 1 because they include the same corresponding limitations of claim 1.

But Darlet does not disclose supporting mutually different categories of instruction types; nor does Darlet disclose grouping the memory locations, each according to the category of instructions; nor storing the symbolic reference result for an instruction of the category to which a group is assigned. Nor does Darlet disclose executing the instruction such that if there no association symbol reference/result in the above categories-based group in memory locations, resolving the particular reference, using the result from such resolving as operand data for executing the instruction and storing an association symbolic reference/ result of said resolving step in the group assigned to the category of the particular instruction.

But these limitations have been addressed correspondingly in claim 1 using Gee and Lai.

As per claim 9, refer to claim 2.

Conclusion

5. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Pat No. 6,499, 095 to Sexton et al., disclosing saving numeric reference in loading of objects.

U.S. Pat No. 6,446, 084 to Shaylor et al.,disclosing resolution type reference based on class, method, interface.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (703) 305-7207. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662.

Any response to this action should be mailed to:

Commissioner of Patents and Trademarks

Washington, D.C. 20231

or faxed to:

(703) 872-9306 (for formal communications intended for entry)

or: (703) 746-8734 (for informal or draft communications, please label

"PROPOSED" or "DRAFT" – please consult Examiner before use)

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington. VA. , 22202. 4th Floor(Receptionist).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

VAT
June 1, 2004

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100